

プロンプトチューニングと kNN-LM を組み合わせた リスティング広告のタイトル自動生成

児玉壮平¹ 星野智紀² 石塚湖太²¹ 東京理科大学大学院 ² 株式会社博報堂テクノロジーズ¹8623506@ed.tus.ac.jp²{tomoki.hoshino,kota.ishizuka}@hakuodo-technologies.co.jp

概要

リスティング広告のタイトル自動生成には、Fine-Tuning された事前学習済み言語モデルの活用が盛んである。また、Fine-Tuning はモデルやデータが大規模になると計算コストがより増加する。本研究では、Instruction-Tuning 済みモデルに kNN-LM を導入し、プロンプトチューニングと用例検索によって広告タイトルを生成する。さらに、kNN-LM において用例検索を行うモデルと生成を行うモデルを分けることで、プロンプトチューニングの度に kNN-LM のデータストアを再構築する必要をなくす。実験によって、提案手法によって広告タイトルの生成品質が向上することを確認し、計算コストを削減できることを確認した。

1 はじめに

リスティング広告とは、ランディングページ（以下、LP）の広告をユーザーが検索した検索クエリに関連して表示する広告手法である。近年、検索クエリと LP の HTML タグ情報を含んだ入力を言語モデル（以下、LM）に与えて、リスティング広告の広告タイトルを自動生成させることが盛んである [1]。

現在では、事前学習済み LM に Fine-Tuning を施すことで、広告タイトルを生成することが一般的である。一方、広告タイトルの文字数制限などの規定は掲載するプラットフォームによって変わり、また、社会のトレンドに応じて広告のあり方が変わる。そのため、その度に Fine-Tuning が必要になるが、Fine-Tuning は LM が大規模になると計算コストが増加するため、広告の規定やトレンドへの対応が困難になる。

本研究では、Instruction-Tuning [2] 済み LM を使用し、さらに kNN-LM [3] を導入して広告タイト

ル生成を行う手法を提案する。Instruction-Tuning 済み LM に対してプロンプトチューニングを行うことで、広告の規定やトレンドに対応する。また、kNN-LM は、入力に対する予測トークンのデータストアを事前に作成し、そのデータストアに対して用例検索を行うことで、Fine-Tuning なしで生成品質を向上させる。

さらに、kNN-LM において、データストアを構築して用例検索をする LM（以下、 LM_{kNN} ）と生成を行う LM（以下、 LM_{gen} ）を分ける（図 1）。従来の kNN-LM は、 LM_{kNN} と LM_{gen} は同じものを用い、入力も同じである。そのため、プロンプトチューニングによって入力を変えた場合、その度にデータストアを作り直す必要がある。そこで、 LM_{kNN} と LM_{gen} を分けることによって、データストアと LM_{kNN} を固定しつつ、 LM_{gen} に対してプロンプトチューニングを行うことを可能としている。実験から、提案手法によって Instruction-Tuning 済み LM より最大で ROUGE-1 は 0.117、ROUGE-2 は 0.107 改善したことや、 LM_{kNN} と LM_{gen} で異なる LM を使うことによって性能の低下がないことを確認し、データストアを作り直す計算コストを削減できることを確認した。

2 kNN-LM

kNN-LM では、主に「データストア構築」と「推論時」の二つの段階に分けられる。

2.1 データストア構築

通常の文生成では、LM は時刻 t 以前のトークン列 $\mathbf{c}_{<t} = (w_1, \dots, w_{t-1})$ が与えられた場合、次トークン w_t の分布 $P(w_t | \mathbf{c}_{<t})$ を推定する [4]。データストア $(\mathcal{X}, \mathcal{Y})$ には、 $(\mathbf{c}_{<t}, w_t) \in \mathcal{D}$ が与えられたとき、 $\mathbf{c}_{<t}$ を LM に入力した際に得られるデコーダ最終層

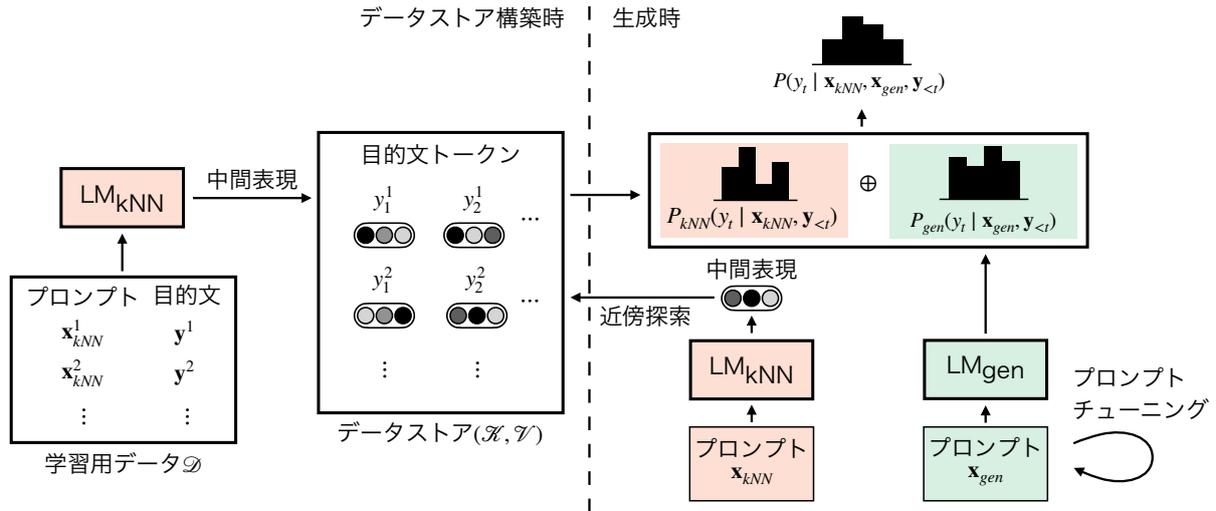


図 1: 自動生成モデルの概要

の中間表現をキーとし、次トークン w_t を値として、

$$(\mathcal{X}, \mathcal{V}) = \{(f(\mathbf{c}_{<t}), w_t) \mid (\mathbf{c}_{<t}, w_t) \in \mathcal{D}\} \quad (1)$$

と格納する。ここで、 \mathcal{D} は訓練データ、 $f(\cdot)$ は入力から LM のデコーダ最終層の中間表現を得る関数である。本研究では、Khandelwal ら [3] に倣い、フィードフォワード層の入力ベクトルを中間表現として用いている。

2.2 生成

生成時には、入力のトークン列 $\mathbf{c}_{<t}$ が与えられた際、LM は次トークンの予測出現確率 $P_{gen}(w_t \mid \mathbf{c}_{<t})$ と中間表現 $f(\mathbf{c}_{<t})$ を出力する。この $f(\mathbf{c}_{<t})$ を検索クエリとしてデータストアに近傍探索を行い、距離関数 $d(\cdot, \cdot)$ に基づいて k 近傍 \mathcal{N} を取得する。その後、負の距離の softmax に基づいて各近傍に対する予測出現確率を

$$P_{kNN}(w_t = y \mid \mathbf{c}_{<t}) \propto \sum_{(k_i, v_i) \in \mathcal{N}} \mathbb{1}_{y=v_i} \exp(-d(k_i, f(\mathbf{c}_{<t}))) \quad (2)$$

と計算し、 \mathcal{N} の中での各語彙に集約する。ここで、 \mathcal{N} に現れない語彙の予測出現確率は 0 となる。本研究では、Khandelwal らに倣い、距離関数は L2 距離を用いている。最終的な次トークンの予測出現確率は P_{kNN} と P_{gen} の線形補間によって、

$$P(w_t \mid \mathbf{c}_{<t}) = \lambda P_{kNN}(w_t \mid \mathbf{c}_{<t}) + (1 - \lambda) P_{gen}(w_t \mid \mathbf{c}_{<t}) \quad (3)$$

と計算する。ここで、 λ はハイパーパラメータである。

3 提案手法

3.1 概要

本提案手法では広告タイトル自動生成モデルとして、kNN-LM を導入した Instruction-Tuning 済み LM を使い、 LM_{kNN} と LM_{gen} を分けることによって、プロンプトチューニングの度にデータストアを再構築する必要をなくす (図 1)。

3.2 データストア構築

提案手法では、LP の情報をプロンプトとして与えて、広告タイトルを生成させるが、プロンプト部分は用例検索をする必要がない。そのため、提案手法におけるデータストア構築の際には、プロンプト部分は格納せず、広告タイトルのみ格納する。プロンプト $\mathbf{x}_{kNN} = (x_1, x_2, \dots, x_{|\mathbf{x}_{kNN}|})$ と目的広告タイトル $\mathbf{y} = (y_1, y_2, \dots, y_{|\mathbf{y}|})$ が与えられたとき、時刻 t には、プロンプト \mathbf{x}_{kNN} と時刻 t までの目的広告タイトル $\mathbf{y}_{<t}$ をモデルに入力した際に得られる中間表現 $f(\mathbf{x}_{kNN}, \mathbf{y}_{<t})$ をキー、 \mathbf{y}_t を値として、

$$(\mathcal{X}, \mathcal{V}) = \{(f(\mathbf{x}_{kNN}, \mathbf{y}_{<t}), \mathbf{y}_t) \mid (\mathbf{x}_{kNN}, \mathbf{y}) \in \mathcal{D}\} \quad (4)$$

をデータストアに格納する。

3.3 生成

提案手法では LM_{kNN} と LM_{gen} と分けているため、それぞれに異なるプロンプトを入力することが可能となる。 LM_{kNN} に入力するプロンプトを \mathbf{x}_{kNN} 、 LM_{gen} に入力するプロンプトを \mathbf{x}_{gen} とするとき、次

表 1: 広告文自動生成の結果

HTML タグ	\mathbf{x}_{gen}	Metrics	P_{gen}	$LM_{kNN-\mathbf{x}_{kNN}}$			
				I-F	I-Z	P-F	P-Z
全て	Few	R-1	0.278	0.343	0.353	0.338	0.347
		R-2	0.127	0.187	0.194	0.181	0.187
	Zero	R-1	0.260	0.340	0.348	0.328	0.338
		R-2	0.110	0.187	0.194	0.177	0.183
タイトル	Few	R-1	0.277	0.317	0.330	0.320	0.334
		R-2	0.118	0.159	0.174	0.164	0.173
	Zero	R-1	0.268	0.327	0.334	0.329	0.340
		R-2	0.109	0.168	0.176	0.169	0.175
見出し	Few	R-1	0.242	0.292	0.316	0.296	0.311
		R-2	0.095	0.141	0.159	0.142	0.154
	Zero	R-1	0.247	0.291	0.314	0.286	0.308
		R-2	0.103	0.141	0.160	0.136	0.153
キーワード	Few	R-1	0.236	0.311	0.335	0.319	0.331
		R-2	0.079	0.154	0.175	0.157	0.170
	Zero	R-1	0.216	0.310	0.333	0.314	0.328
		R-2	0.068	0.159	0.175	0.159	0.168

トークンの予測出現確率は

$$P(y_t | \mathbf{x}_{kNN}, \mathbf{x}_{gen}, \mathbf{y}_{<t})$$

$$= \lambda P_{kNN}(y_t | \mathbf{x}_{kNN}, \mathbf{y}_{<t}) + (1 - \lambda) P_{gen}(y_t | \mathbf{x}_{gen}, \mathbf{y}_{<t}) \quad (5)$$

と計算される。

4 実験

本実験では、従来法と提案法の kNN-LM を用いた広告タイトル生成の性能の差異を検証するため、それぞれの手法で広告タイトルを生成し、その品質を比較する。LM_{gen} は Instruction-Tuning 済み LM で固定し、LM_{kNN} は Instruction-Tuning されていない事前学習済み LM を使用した場合と Instruction-Tuning 済み LM を使用した場合の二つの条件で変更した。また、プロンプトチューニングが及ぼす影響を検証するため、 \mathbf{x}_{gen} と \mathbf{x}_{kNN} それぞれでプロンプトを Few-shot と Zero-shot の両方の条件で変更して実験を行った。更に、実験結果の信頼性を確認するため、使用するタグ情報を異なるパターンに分けて検証した。

4.1 使用データ

使用するデータセットは、検索クエリ、ランディングページ (LP) と広告タイトルの関係を学習するために、独自にスクレイピングすることで得られた

表 2: P_{kNN} のみによる自動生成の結果

HTML タグ	Metrics	$LM_{kNN-\mathbf{x}_{kNN}}$			
		I-F	I-Z	P-F	P-Z
全て	R-1	0.326	0.342	0.312	0.332
	R-2	0.175	0.187	0.164	0.179
タイトル	R-1	0.306	0.326	0.311	0.326
	R-2	0.152	0.171	0.156	0.172
見出し	R-1	0.274	0.308	0.281	0.304
	R-2	0.130	0.151	0.134	0.147
キーワード	R-1	0.303	0.332	0.309	0.325
	R-2	0.152	0.175	0.155	0.168

データである。各データは検索クエリ、リスティング広告で表示された広告タイトル、LP の HTML から構成されている。データは約 45 万件あり、そのうち 99% を訓練用、1% を評価用として使用し、それぞれの内 10% を Few-Shot の例として使用している。

4.2 評価方法

生成品質は参照文との類似性を評価する手法である ROUGE[5] で評価する。ROUGE は ROUGE-1 と ROUGE-2 (以下, R-1, R-2) の二通りで評価した。

5 結果と考察

5.1 結果

4 章の実験の結果を表 1 に示す。ここで、 P_{gen} は LM_{gen} のみによる生成の精度を表し、LM_{kNN- \mathbf{x}_{kNN}} はそれぞれのイニシャルで表現している (I-F は Instruction-Tuning 済み LM に Few-shot のプロンプトを、I-Z は Instruction-Tuning 済み LM に Zero-shot のプロンプトを、P-F は事前学習済み LM に Few-shot のプロンプトを、P-Z は事前学習済み LM に Zero-shot のプロンプトを与えた場合を表す)。また、(5) 式の λ は 0.1 から 0.9 まで 0.1 刻みの間隔で変動させ、最も R-1 が良い値の結果を表に記載している。同じ HTML タグと \mathbf{x}_{gen} を用いた場合で比較したとき、 P_{gen} のみによる生成より従来法の kNN-LM (背景色赤) の方が最大で R-1 は 0.117, R-2 は 0.107 改善している。

また、 \mathbf{x}_{kNN} を変更した場合 (背景色緑) は、従来法の kNN-LM (背景色赤) に比べ、最大で R-1 は 0.023, R-2 は 0.019 の劣化に留まっている。LM_{kNN} を事前学習済み LM とし \mathbf{x}_{kNN} を Zero-shot とした場合 (背景色青) は、従来法 (背景色赤) に比べ、最

表 3: 各 \mathbf{x}_{gen} と $\text{LM}_{\text{kNN}}\text{-}\mathbf{x}_{\text{kNN}}$ の組み合わせにおける生成例

\mathbf{x}_{gen}	$\text{LM}_{\text{kNN}}\text{-}\mathbf{x}_{\text{kNN}}$	生成タイトル
Few	-	【公式】 Commerce Flow
	I-F	Amazon 広告を AI が自動で改善
	I-Z	Amazon 広告を自動化して、売上アップ
	P-F	Amazon 広告 パートナーにご相談ください
	P-Z	Commerce Flow で Amazon の広告を最適化
Zero	-	Commerce Flow, Amazon, パートナー, 広告, AI, Amazon 広告
	I-F	Amazon で売る方法、これでもう迷わない
	I-Z	Amazon で売る個人事業者のための広告運用ツール
	P-F	Commerce Flow(Amazon 広告自動運用ツール)
	P-Z	Amazon で売るなら Commerce Flow

大で R-1 は 0.010, R-2 は 0.011 の劣化となっている。

P_{kNN} のみによる自動生成の結果を表 2 に示す。同じ HTML タグと LM_{kNN} を用いた場合、 \mathbf{x}_{kNN} が Few-shot よりも Zero-shot の方が優位であり、最大で R-1 が 0.034, R-2 が 0.023 改善している。

また、『Amazon 広告自動運用ツール「Commerce Flow (コマースフロー)」』という Amazon 広告の自動運用サービスの LP¹⁾ に対して、実際に生成した広告タイトルを表 3 に示す。ここで、生成例は HTML タグを全て用いた場合を記載している。

5.2 考察

表 1 の結果より、 P_{gen} のみによる生成より従来法の kNN-LM による生成の方が優位であるといえる。これは、用例検索によって広告タイトルでよく用いられる文体を生成することが可能になったからだと考えられる。また、従来法の kNN-LM と、 \mathbf{x}_{kNN} を変更した場合や LM_{kNN} を事前学習済み LM に変更した場合を比較したとき、精度の劣化が最大でも R-1 が 0.023, R-2 が 0.019 であったことから、kNN-LM においてモデルを分けることによる影響は小さい。

表 2 の結果より、 P_{kNN} は \mathbf{x}_{kNN} が Zero-shot の方が性能が高いと言える。データストアに対する生成時の用例検索の際、例として用いたプロンプトと広告タイトルの情報を含んだベクトルがクエリとなるため、目的となる広告タイトルとは関係のない情報を含んだ上での推論を行う。これが、Zero-shot の方が優位である理由であると考えられる。

次に、実際の生成例に注目する。 \mathbf{x}_{gen} が Few-shot の場合、従来法の kNN-LM (背景色赤) は Instruction-

Tuning 済みモデルのみの生成 ($\text{LM}_{\text{kNN}}\text{-}\mathbf{x}_{\text{kNN}}$ が「-」) よりも、生成される広告タイトルに Amazon などの重要な単語が含まれており、商品の詳細が伝わるタイトルになっている。また、提案手法である LM_{kNN} や \mathbf{x}_{kNN} を変えた場合も大きく生成品質が劣化していないことがわかる。 \mathbf{x}_{gen} が Zero-shot の場合も同様に、Instruction-Tuning 済みモデルのみの生成では単語の羅列であった生成タイトルが、kNN-LM を導入することで改善されていることが分かる。

以上のことから、プロンプトをある LM に与えた際の中間表現を用いてデータストアを構築し、 LM_{gen} と LM_{kNN} でモデルを分け、 LM_{gen} に対してプロンプトチューニングを行うことで、広告の規定やトレンドに対応することが可能であると言える。

6 まとめ

本研究では、広告タイトル自動生成モデルとして、kNN-LM を導入した Instruction-Tuning 済み LM を用い、 LM_{kNN} と LM_{gen} を分ける手法を提案した。実験の結果、 LM_{kNN} と LM_{gen} を分けることによる悪影響はなく、むしろ性能が上がる可能性があることを確認した。これにより、プロンプトチューニングの度にデータストアを構築する必要がなく、より迅速に広告タイトルの形式的要素のトレンドに対応することが可能となる。今後の展望として、Fine-Tuning された LM との精度の比較することや、人手評価による広告効果の検証をすることなどが挙げられる。

1) <https://commerce-flow.com/>

謝辞

本研究は、株式会社博報堂テクノロジーズのインターンシップとして実施した。

参考文献

- [1] 村上聡一朗, 星野翔, 張培楠. 広告文自動生成に関する最近の研究動向. 人工知能学会全国大会論文集, Vol. JSAI2022, No. 1P5GS601, 2022.
- [2] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In **International Conference on Learning Representations**, 2022.
- [3] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In **International Conference on Learning Representations**, 2020.
- [4] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [5] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [6] Khalid Alnajjar and Hannu Toivonen. Computational generation of slogans. **Natural Language Engineering**, Vol. 27, No. 5, pp. 575–607, 2021.
- [7] Stamatina Thomaidou, Ismini Lourentzou, Panagiotis Katsivelis-Perakis, and Michalis Vazirgiannis. Automated snippet generation for online advertising. In **Proceedings of the 22nd ACM International Conference on Information & Knowledge Management**, CIKM '13, pp. 1841–1844, New York, NY, USA, 2013. Association for Computing Machinery.
- [8] Kevin Bartz, Cory Barr, and Adil Aijaz. Natural language generation for sponsored-search advertisements. In **Proceedings of the 9th ACM Conference on Electronic Commerce**, EC '08, pp. 1–9, New York, NY, USA, 2008. Association for Computing Machinery.
- [9] 大曾根宏幸, 張培楠. Gpt-2 の転移学習によるキーワードを考慮した広告文生成. 人工知能学会全国大会論文集, Vol. JSAI2021, No. 2D4OS7b03, 2021.
- [10] 黒木開, 石塚湖太, 川上孝介, 中田和秀. 複数の指定語句を必ず含むリスティング広告の自動生成. 言語処理学会年次大会発表論文集, Vol. 28, pp. 1339–1343, Mar 2022.
- [11] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 38–45, Online, October 2020. Association for Computational Linguistics.
- [12] Uri Alon, Frank Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. Neuro-symbolic language modeling with automaton-augmented retrieval. In **International Conference on Machine Learning**, pp. 468–485. PMLR, 2022.
- [13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2019.

表 4: 各モデルの入力テンプレート

モデル	テンプレート
事前学習済み LM	「…」をキーワードとして持ち、「…」を見出しとして持つ、「…」という名前のページの、「…」という検索ワードに対する広告効果が高い広告タイトルは「
Instruction-Tuning 済み LM	ユーザー:「…」という名前のランディングページの、「…」という検索ワードに対する広告効果が高い広告タイトルを 30 文字以内で 1 つ出力してください. システム:「

A 関連研究

A.1 広告文自動生成

近年のインターネット広告市場の急速な規模拡大によって、広告文自動生成技術の需要が高まり、広告文自動生成に関する研究が盛んになっている [1]. 広告文自動生成の対象は大きく分けて、リスティング広告、キャッチコピーの 2 つがある. リスティング広告とは、ユーザーが検索した検索クエリに関連したランディングページの広告を検索結果に表示する広告である. キャッチコピーは、商品やサービスに対して、顧客の興味を惹きつけるために用いられる.

テンプレートを用いた広告文自動生成は、人手で作成したテンプレートを用いるため、文法的に正確に魅力的な広告文を生成しやすくなる. Alnajjar らは、キャッチコピー自動生成において、既存のキャッチコピーの内容語等を空欄化したテンプレートを用いている [6]. また、テンプレートに挿入するキーワードを、n-gram の自己相互情報量に基づいて LP から重要なキーワードを抽出する手法や [7], テンプレートと挿入するキーワードの書体を揃えたり、修飾語を追加したりすることによって適切にキーワードを挿入する手法 [8] が提案されている.

最近の広告文自動生成では、大規模なテキストコーパスで事前学習された言語モデルの活用が進んでいる. 大曾根ら [9] は、事前学習された GPT-2 に対して広告文自動生成用に Fine-Tuning することで、広告文自動生成が実現できることを報告している. さらに、BERT2BERT を用いた複数の指定語句を必ず含む手法などが提案されている [10].

B プロンプトチューニング

表 4 に本実験で使用した入力のテンプレートを示す. 事前学習済み LM の場合には、文の続きを生成させるテンプレートを用いる. また、Instruction-Tuning 済み LM の場合には、「ユーザー:」の後に指示を記載し、「システム:」の後に応答を生成させるテンプレートを用いる. プロンプトが Few-shot の場合には、例を 2 つ前に記載する.

C 実験の詳細設定

C.1 計算機

実験における計算機は、NVIDIA DGX POD を 1 基使用した. NVIDIA DGX POD には、CPU として AMD EPYC 7742 @ 2.25GHz (64 コア) が 256 基搭載されており、メモリは 2TB で、OS は Ubuntu 20.04 である. また、GPU として NVIDIA A100 80GB を 8 基搭載している.

C.2 実装

生成に用いる LM として Hugging Face Transformers [11] にて LINE ヤフー社が公開している 3.6B の事前学習済みモデルと Instruction-Tuning 済み LM²⁾ を用いた. kNN-LM の実装には kNN-Transformers [12] を使用し、従来法と提案法の近傍探索には FAISS [13] を使用した.

C.3 パラメータ

近傍探索数は、従来法と提案法共に $k=1024$ としている. その他のパラメータは kNN-Transformers [12] に準拠している.

2) <https://huggingface.co/line-corporation>